

**Technical Paper
November 1999**

**IMPLEMENTING
the
SPATIAL DATA STANDARDS FOR FACILITIES,
INFRASTRUCTURE, AND ENVIRONMENT (SDS)
and
FACILITY MANAGEMENT STANDARDS FOR
FACILITIES, INFRASTRUCTURE, AND ENVIRONMENT
(FMS)
Using
ENVIRONMENTAL SYSTEMS RESEARCH
INSTITUTE, INC'S
ArcInfo version 7.2.1**

Final Draft

**Published by: The CADD/GIS Technology Center for Facilities, Infrastructure, and
Environment
Army Engineer Research and Development Center (at Vicksburg)
3909 Halls Ferry Road
Vicksburg, MS 39180-6199**

ArcInfo Characteristics and Capabilities

ArcInfo is a geographic information system (GIS) developed by Environmental Systems Research Institute, Inc. (ESRI), Redlands, CA. ESRI defines GIS as an organized collection of computer hardware, software, geographic data, and personnel designed to efficiently capture, store, update, manipulate, analyze, and display all forms of geographically referenced “information.” Its first installation was in 1982. The current estimated installation base is over 20,000 with a user community of over 40,000.

ArcInfo contains tools for the capture, correction, management, query, analysis, and display of geographic information. It uses a relational data base model to store and manage both the spatial and descriptive data associated with geographic phenomena. Spatial data can be defined as geographic features with real world locations (e.g., pipeline shutoff valve, oil pipeline, and soil boundary). Attribute data can be defined as descriptions of the above mentioned geographic features (e.g., type of valve, diameter of pipeline, and soil type). The ARC software specifically handles the spatial elements of the GIS. It implements a vector-based geographic data model where geographic features are stored as polygons, arcs, and points called “coverages.” Topology, a method used to define the spatial relationships among features, connects the various entities within a coverage and permits the complex queries and analyses that are inherent to GIS. INFO is the tabular database management system (DBMS) used by ArcInfo to store and manipulate feature attribute and related tables. Simply put, INFO handles feature descriptions and how features relate to each other. A DBMS is a system to provide a means of storing and updating information.

The “geo-relational” model used in vector products like ESRI’s ArcInfo handles spatial and attribute information separately and links them with a unique ID, or relation. The equivalent approach, based on the grid-cell tessellation, is called the “map composite” approach in raster systems. These “hybrid” approaches handle spatial and aspatial well provided the database is structured correctly. The hybrid systems generally include “integrators,” like ArcInfo’s Database Integrator (DBI). The key linkage between spatial and aspatial data in these systems is usually a pointer, or unique ID in vector systems, and the individual cells, or the cell-id, in raster systems.

Additional modules are available in ArcInfo that allow the user to perform other types of spatial data processing. These modules include GRID, NETWORK, TIN, and COGO. For more information on ArcInfo and it’s modules please read the online help or visit the ESRI website at <http://www.esri.com/software/arcinfo/index.html>

Implementation of the SDS/FMS with ESRI's ArcInfo

ArcInfo Software

ArcInfo version 7.2.1 for Windows NT 4.0 was used in the preparation of this report.

Windows NT

Microsoft Windows NT version 4.0 was used as the operating system platform for the preparation of this report.

Relational Databases

Database Management Systems (DBMS) are used to organize and structure data so that it can be retrieved and manipulated by users and application programs. These DBMS', popularized in the '70s and '80s, were later streamlined into what is know today as a Relational Database Management System (RDBMS). The RDBMS is the attempt to simplify the database structure. It eliminates the parent/child structures from the database and replaces it with a simple row or columnar data structures. Simply put, "a relational database is a database where all data visible to the user is organized strictly as tables of data values, and where all database operations work on these tables."¹

The Structured Query Language (SQL) is the standard computer database language. The SDS Generator provided with the CD release of version 1.90 of the SDS/FMS produces SQL statements that can be used to populate blank tables for various types of RDBMS'. These following formats are:

- Standard ANSI
- Microsoft SQL
- Informix
- Oracle
- Microsoft Access

The American National Standards Institute (ANSI) has established the SQL as the standard language for relational databases. The SQL will work with any ANSI compliant SQL software. The other SQL manipulation software such as Informix, Microsoft SQL and Oracle handle SQL statements differently but work in the same general manner using the same standard ANSI SQL format. The SDS/FMS Generator will also produce the Microsoft Access database with the appropriate blank databases. The Oracle method will be accomplished later in this report.

¹ Groff & Weinberg p. 49-70.

Oracle

Oracle7 Server Release 7.3.2 for Windows NT 4.0 was used as the DBMS for the preparation of this report. In addition SQL Plus 3.3 was used as the SQL query software.

Sample Data Set

The data used for the SDS/FMS Implementation Study was taken as a subset of data that was developed for the SDS/FMS training classes conducted at the CADD/GIS Technology Center for Facilities, Infrastructure and Environment in Vicksburg, MS.

SDS/FMS Implementation Scenarios

Different scenarios require specific implementation strategies depending on the data type (graphic or attribute) and format (hardcopy or digital). Those that require consideration are: (1) source documentation for graphic and attribute information that may exist but is not in digital form, (2) graphic data that has yet to be digitized, but contains related attribute values; (3) graphic data that has been created without attribute data having been entered, or (4) graphic and attribute information that has been digitized and entered. To implement any of the above scenarios, a user must have a clear understanding of ArcInfo and the SDS/FMS.

ArcInfo Project Setup with the SDS/FMS

For the following discussion, the applicable implementation scenario is that both graphic and attribute information have been digitized and entered. The graphics were delivered as an ArcView shapefile and the attribute information was delivered as an ASCII text file.

Relational databases, such as ORACLE or SQLServer, are often used to store attribute information for spatially related features. In ArcInfo, the Database Integrator (DBI) provides a generic interface to data stored in a database.

The fundamental purpose of a DBMS connection is to allow related tables to be accessed and updated. A good database design, such as that employed by the SDS/FMS, contains tables (rows or records and columns or fields) of information where each record represents an object and each field represents an object attribute such that every field in every record contains precisely one value - this is called normalization. Relations are used in ArcInfo to temporarily link an INFO table and a related DBMS table using some key field. The related DBMS table can be accessed just as if it were located in INFO.

For the purposes of this guidance, the shapefile was converted to an ArcInfo coverage and the ASCII text file was imported into an Oracle database table.

Database Integrator

Database Integrator is a set of commands within the core ArcInfo product that allows interactive and programmatic access to relational database tables and functions using standard SQL statements. There are three basic commands to access tables connected through Database Integrator: DBMSEXECUTE, DBMSCURSOR, and DBMSINFO/INFODBMS.

DBMSEXECUTE allows the user to issue a standard SQL statement to any connected database and view the results. The SQL statements in Database Integrator must consist of standard SQL commands and must not contain vendor specific SQL extensions. These statements must meet all host database syntax requirements.

DBMSEXECUTE commands can be issued on one ArcInfo command line or by entering a multi-line dialog mode. Using the command line option limits the command line length to 1024 characters, while the multi-line dialog mode allows the user to enter 4000 characters. The DBMSEXECUTE command displays the results of the SQL statement in the window in which the command was issued.

DBMSCURSOR produces an ArcInfo cursor, a pointer to a set of selected rows in a table, from a standard SQL query statement issued to any connected database. Cursor commands allow users to navigate, inspect, or update the selected set of records. The DBMSCURSOR command, like DBMSEXECUTE, uses standard SQL commands properly formatted for a specific host database system and can be issued on a single command line or through the multi-line dialog.

DBMSINFO copies a relational database table or view into an INFO table and automatically converts the database field definitions to types allowable in INFO.

ArcInfo Workspace Setup

In order to organize the data, it is advisable to create a workspace under the user's directory to store all information. This is done by using the ArcInfo command, "CREATEWORKSPACE." Typically, a central GIS database is created along with various user workspaces following a distributed approach.

The SDS/FMS does not specify how to organize the graphic data when the same data type is presented on multiple coverages. However, through the use of ArcInfo's Map Librarian, the Entity Set, Class, and Type can be used to assure unique files. For example, only the coverage names that are defined for SDS/FMS Entity Types are applicable in ArcInfo. Each dataset may require identification in a project specific directory structure and, therefore, the SDS/FMS coverage names will have to be modified. For example, if geology coverages are organized by counties, each county will have a "gesurfea" which will not identify the files uniquely and may cause problems when transferring files under the same name.

The following ArcInfo-specific terms are used to group and organize spatial and non-spatial data relevant to a task:

Geospatial Data-Geographic information in ArcInfo is stored in the system as “coverages.” These coverages represent spatial information and relate to Entity Types in the SDS/FMS. ArcInfo stores geospatial information as point, line, or polygon topology, or in a raster grid structure.

Features - ArcInfo features are spatially distributed geographic elements that compose a coverage.

Attributes - Attributes consist of aspatial information describing the features . ArcInfo attributes are stored in attribute tables with the suffix .AAT (arcs or lines), .PAT (points and polygons), or .VAT (grid values).

Categories -Within ArcInfo, categories are part of map libraries and represent a group of thematically or geographically related coverages.

Central Database-The ArcInfo-ORACLE environment is the combination of geographic information (geographic objects and attribute data) from all sources.

ArcInfo/SDS/FMS Tutorial (A Step-by-Step Procedure)

For this tutorial, data provided by the CADD/GIS Center is utilized. The graphics file (lightpro.shp) and the database file (lights.txt) will be delivered electronically along with this document. In addition, the shapefiles bggen.shp and trveh.shp are supplied for reference information.

Step 1. The first step in the implementation process is to perform an inventory of existing, pertinent data files. A spreadsheet, such as Microsoft Excel provides a relatively simple tool for organizing this inventory.

Step 2. Once an inventory of available data has been made, the next step is to select the features in the SDS that correspond to features in your existing data. The SDS provides a Browser tool to facilitate the feature search process. This tool allows the user to browse by structure, feature, key word, or data source. Listed below is a brief tutorial as to how to use the SDS/FMS version 1.90 browser.

1. Select “SDS/FMS Browser” from the Windows Start Menu



2. Configure the Browser Data Connections

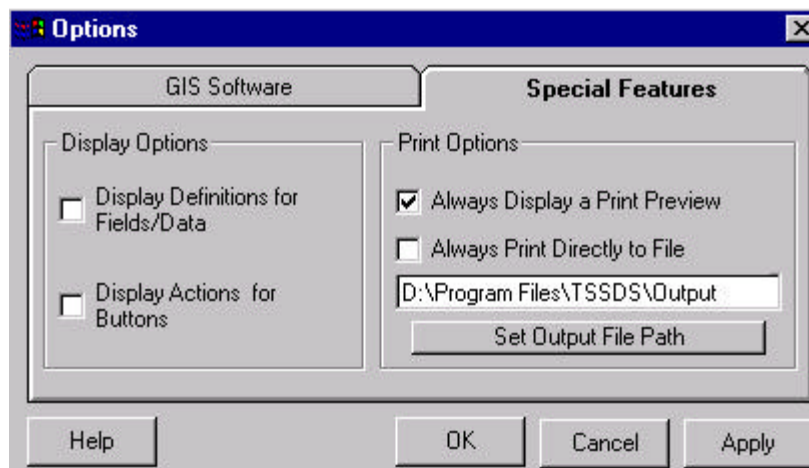
- a. Select “SDS/FMS,” “Connect.”
- b. Verify and change, if necessary, the locations of the SDS/FMS files.

3. Configure the SDS/FMS options

- a. Select “SDS/FMS,” “Options.”
- b. Select the desired GIS software application and select “OK” or “Apply.”



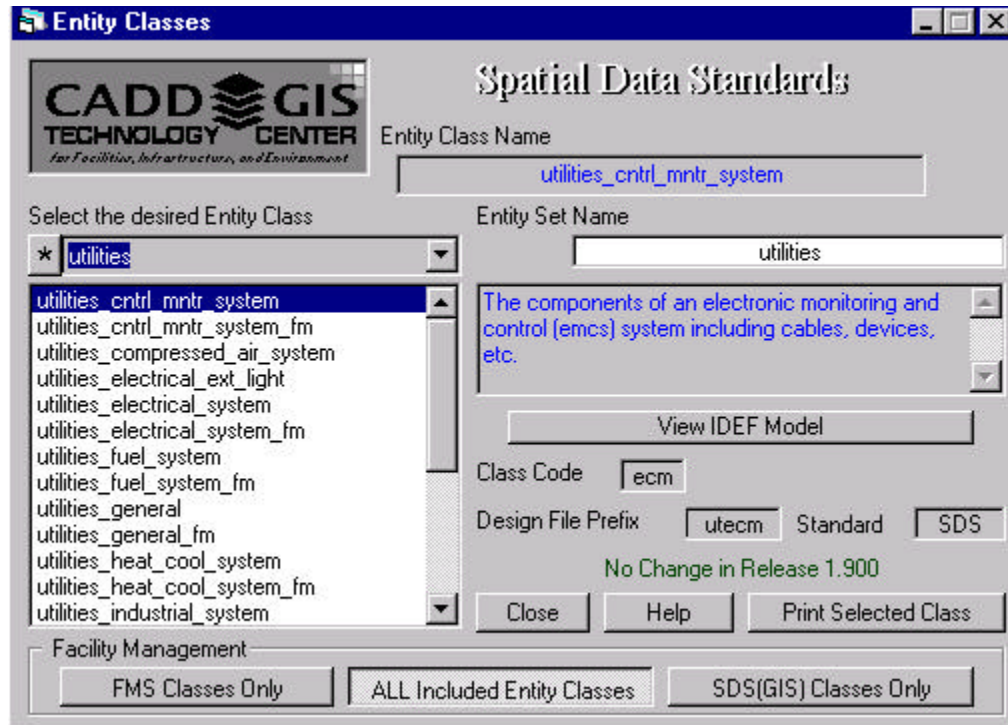
- c. Select the appropriate special features and output file path and select “OK” or “Apply.”



4. Browse the SDS/FMS to determine all the Entity Classes in the “Utilities” Entity Set

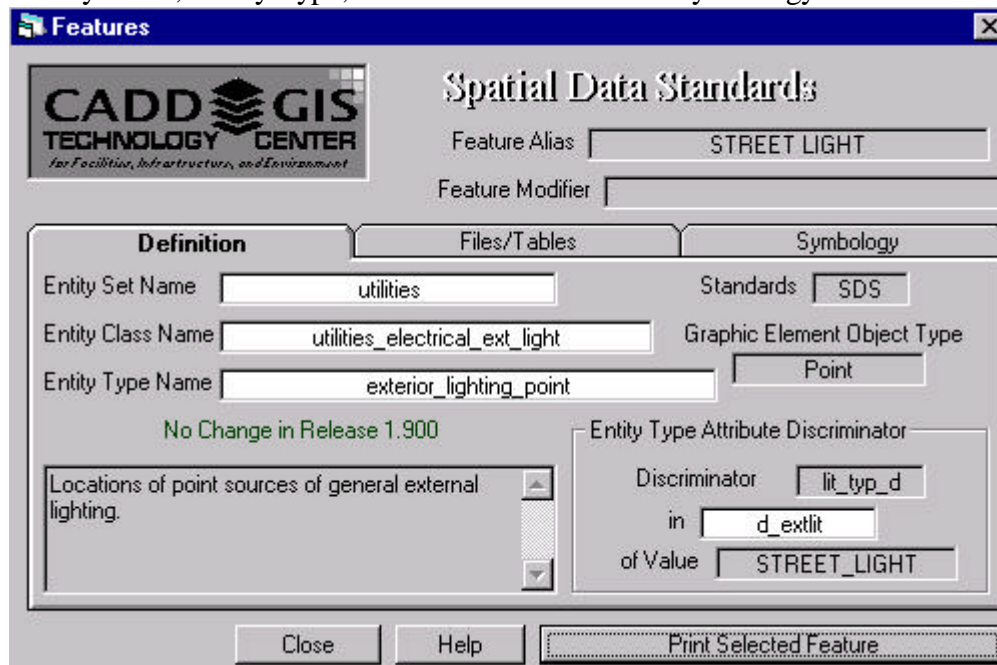
- a. Select “Browse,” “By Structure.”

- b. Click the Entity Classes Bar and change “All Entity Sets” to “utilities” in the Entity Set window.



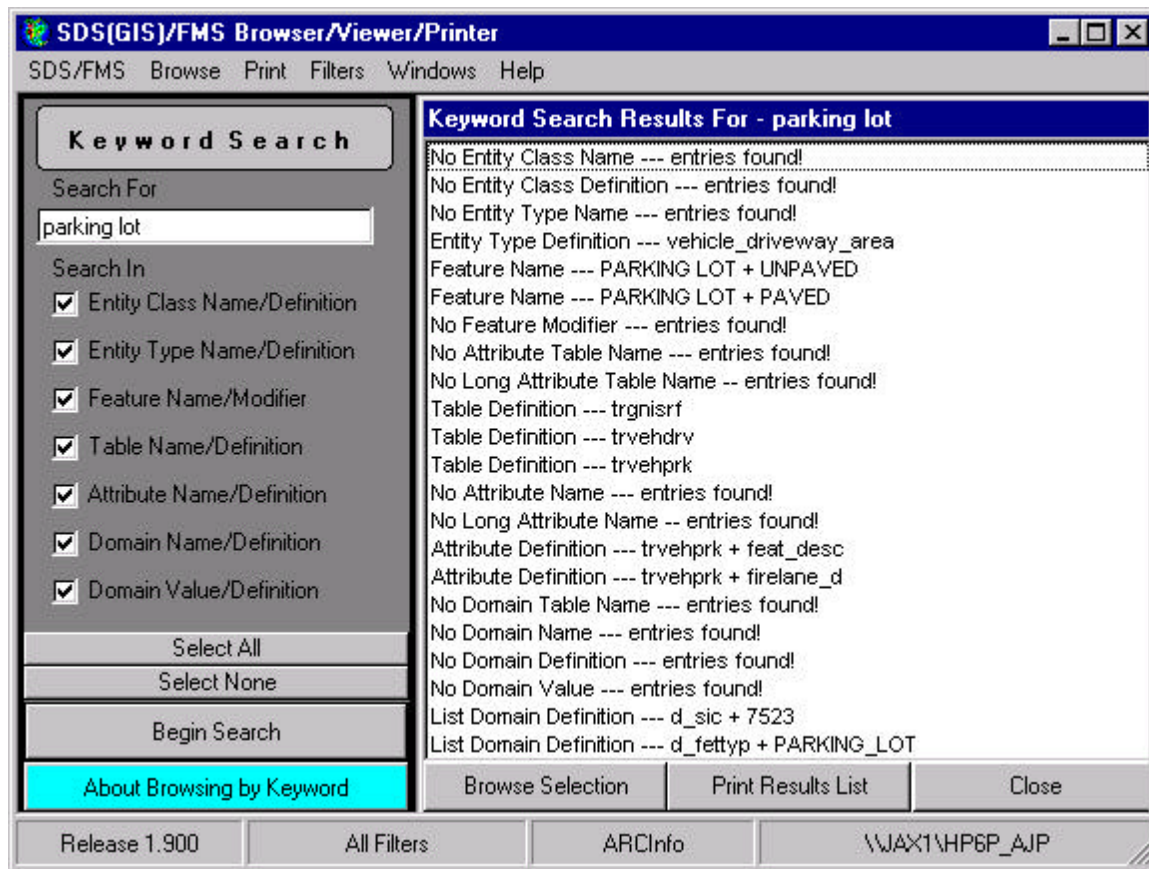
5. Browse the SDS/FMS to determine where Street Lights are defined

- a. Select “Browse,” “By Feature.”
- b. Find “STREET LIGHT.”
- c. Double-click on “STREET LIGHT” to view the SDS/FMS Feature’s Entity Set, Entity Class, Entity Type, attribute table name and symbology.



6. Browse the SDS/FMS to determine where a paved parking lot would be defined

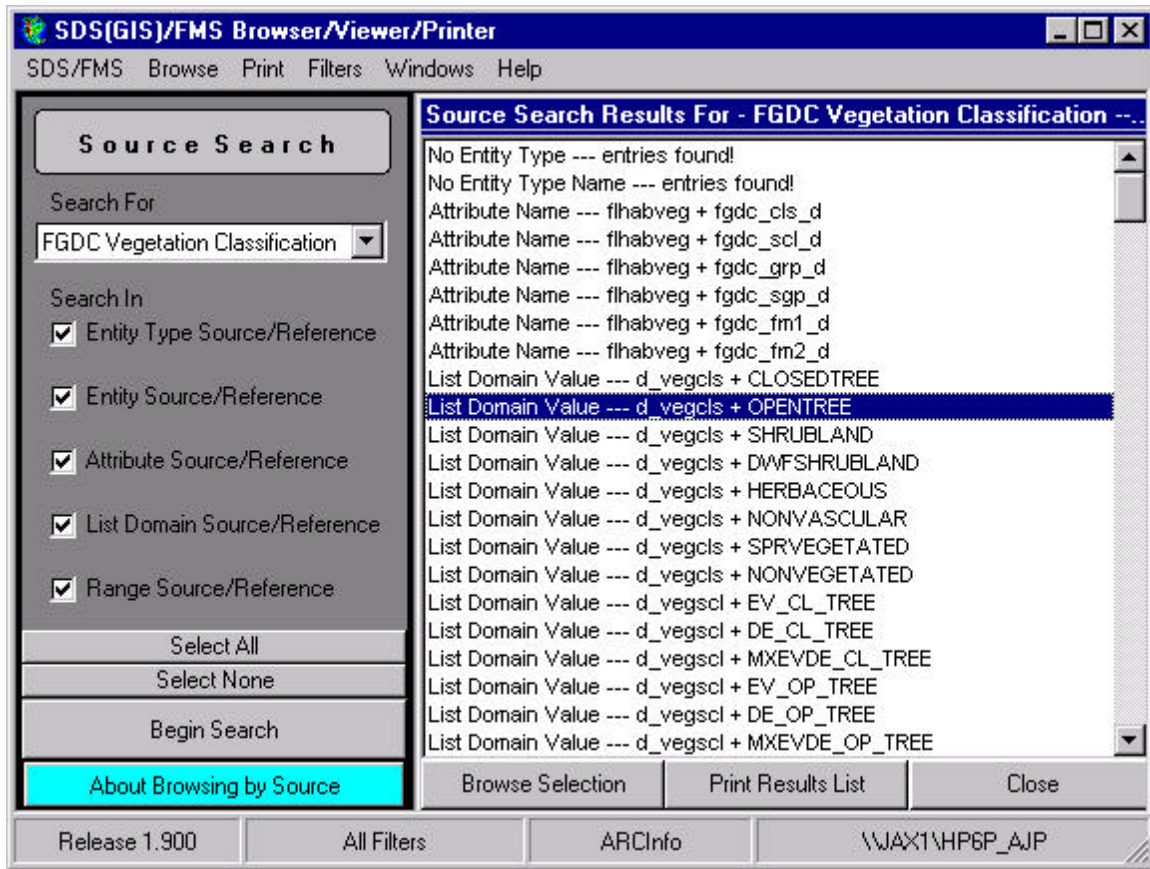
- a. Select “Browse,” “By Keyword.”
- b. Enter “parking lot” into the “Search For” field.



- c. Double-click on Feature name – “PARKING LOT + PAVED” to view the SDS/FMS Paved Parking Lot Feature’s Entity Set, Entity Class, Entity Type, attribute table name and symbology.

7. Browse the SDS/FMS to determine which SDS/FMS domain table contains the FGDC vegetation classification for “opentree”

- a. Select “Browse,” “By Data Source.”
- b. Select “FGDC Vegetation Classification” in the “Search For” field by clicking the down arrow.
- c. Select “Search In” options and then “Begin Search.”



- d. Double-click the List Domain Value –“d_vegcls + opentree” to view the SDS/FMS domain table name, values, and definitions.

Step 3. As you find corresponding SDS/FMS features, it will be helpful to record the entity set, class and type in the spreadsheet file created in step 1 for easy reference during project planning.

Schemas/Filter Creation

Step 4. New to release 1.9 of the SDS/FMS are the Filter Maker and Filter Eraser. The Filter Maker permits the development, definition, modification, implementation, and saving of a single Custom Filter per SDS/FMS Library. A filter is a subset of features used to limit the volume of material used as a part of the SDS/FMS. Filters limit all aspects of the SDS including Entity Sets, Classes, and Types, as well as Tables, Attributes, and Domains. A custom filter is a user-defined filter. Although the creation of a custom filter is optional, it will greatly facilitate the SDS database generation process and is therefore highly recommended.

For the purpose of this tutorial a small filter named “Tutorial” will be built. The following steps will create the filter for purposes of this exercise. *Note: SDS/FMS data does not support filter operations with pre-SDS/FMS 1.8 releases.

1. Engage the SDS/FMSmaker.exe on your SDS/FMS version 1.9 CD.

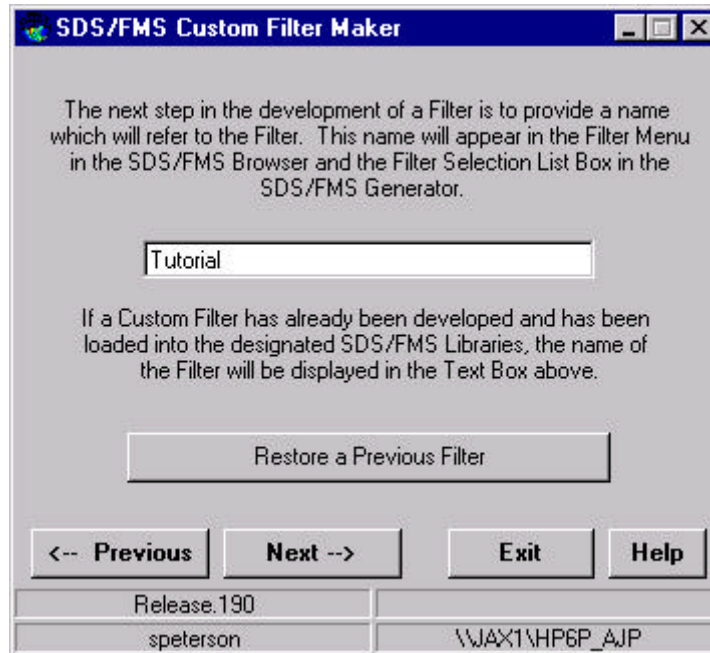
2. Connect to the SDS/FMS Library

- The process is the same as that of the SDS/FMS browser.
Click “Connect to SDS Library”, go to the directory that holds your SDS/FMS browser and choose the “Release.190” directory and click “Connect, Test, and Save”.

Once you connect to the SDS/FMS database, you may proceed to the next step.

3. Naming and/or Retrieving an Existing Filter

- In the box provided to you, type in the word “Tutorial”. Click Next.



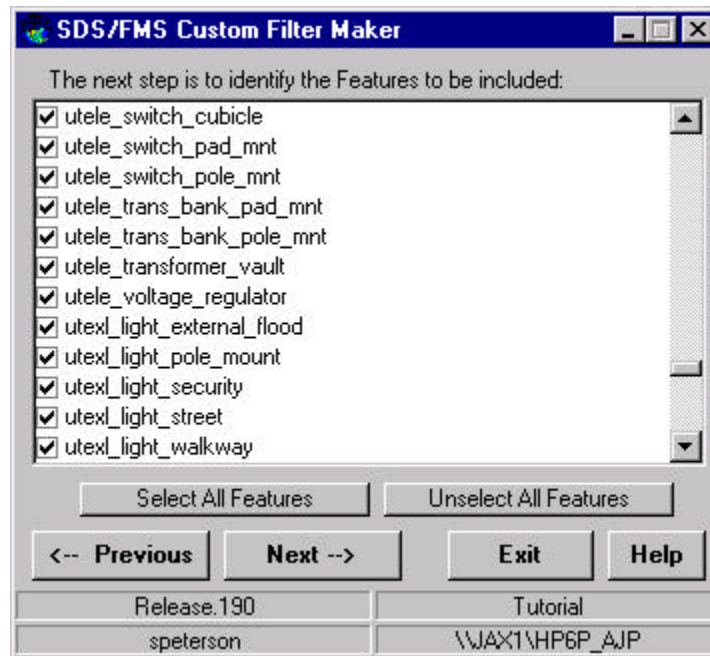
4. Saving the Results for Future Use

- Click the “Save Filter File Name Assignment” button and save tutorial.sdf in an appropriate file directory. (It is suggested that it be saved in the SDS/FMS directory.) Click Next.

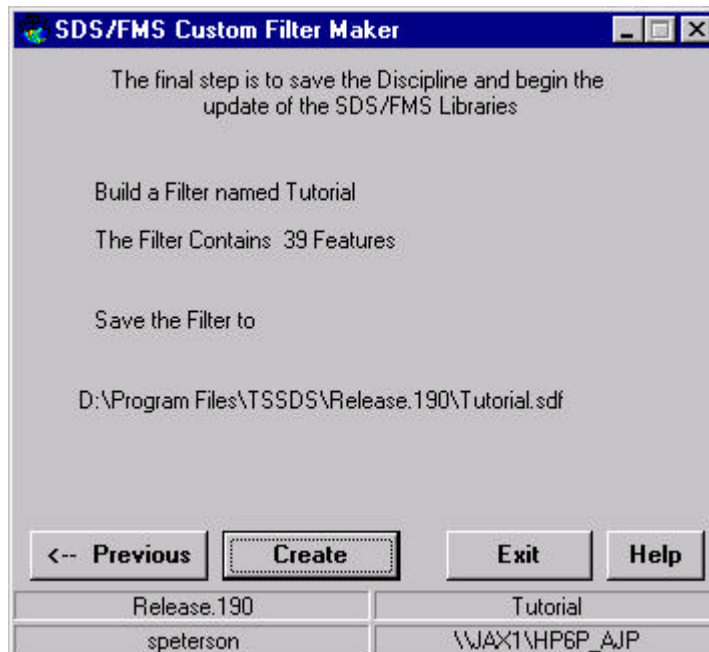
5. Selecting the Features in the Filter

- At this point the filter maker will provide users a list of all SDS/FMS features that are available to put into their custom filter.
- A user has three options here: “Select All Features,” “Unselect All Features,” or check individual features using the check box by the feature names.

- For the purposes of this tutorial select all of the features that begin with “utele” and “utext”. Click Next.



- If the information on the screen is correct, click the Create button. (It will take a while for the filter to be created.) You should see the following
 - Build a Filter named Tutorial
 - The Filter Contains 39 Features
 - Save filter to your chosen directory



- Once the filter generator has completed, open your SDS/FMS 1.9 browser and “Tutorial” should be an available choice from the “Filters” menu.

Erasing a Filter

The following steps are to be used in erasing a user created filter (*Note: the SDS/FMS Filter Eraser is a separate program like the Filter Maker):

- 1. Connect to the SDS/FMS library. Click “Next.”**
- 2. At the Filter Name Screen, observe the name of the Filter; if accurate, click “Next.”**
- 3. If the information displayed is correct, click Erase and click Complete to complete the process.**

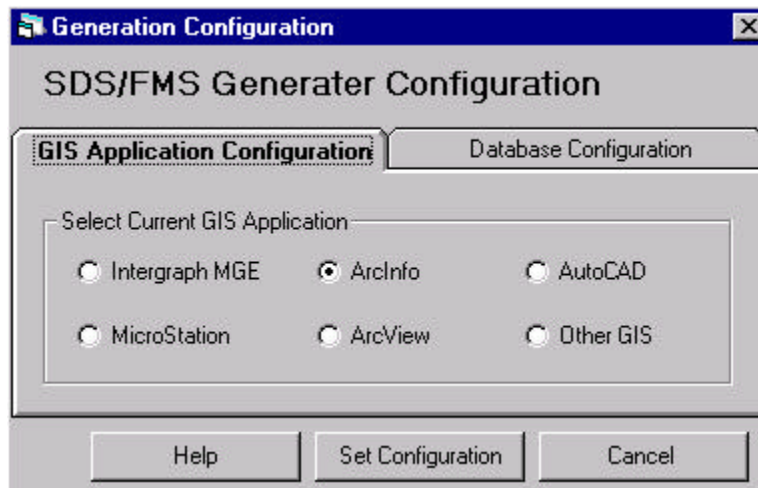
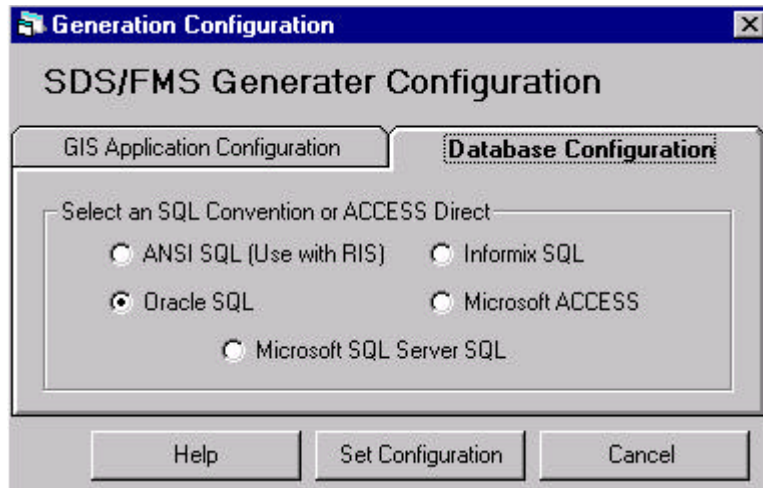
Oracle SQL Generation

Step 5. The next step in the process of ArcInfo implementation involves the use of a new tool in version 1.8 of the SDS/FMS, the SDS Generator. The user has the option to generate database schema in various forms by using the following options:

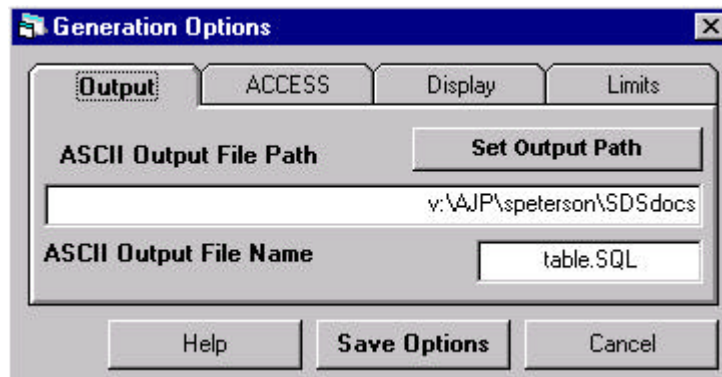
- The entire SDS/FMS schema
- Custom filter schemas
- Single table schemas
- Entity Set schemas
- Entity Class schemas

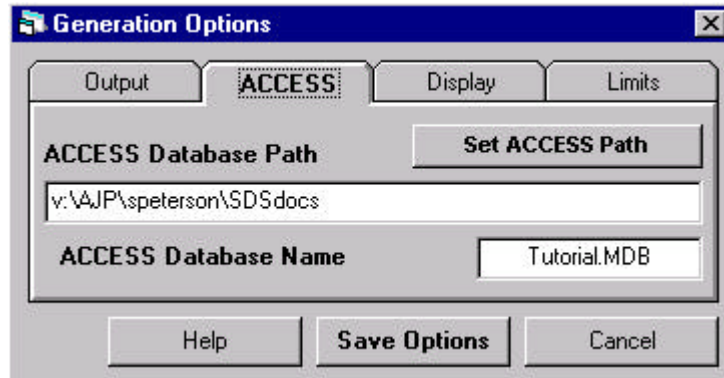
The user also has the choice of creating SQL Script (for Oracle, SQL Server, or Informix) or of creating Access Database tables. For this tutorial the custom filter schema method for creation of Oracle SQL statements will be used. See the online help in the SDS Generator’s help file for more information on the other methods. The following steps will generate the Oracle database tables for the “Tutorial” filter.

- 1. Engage the SDS/FMSgen.exe in the SDS/FMS directory.**
- 2. Connect to the SDS/FMS database.**
- 3. Choose “Configuration” under SDS/FMS and under the Database Configuration tab select “Oracle SQL” and under the GIS Application Configuration select “ArcInfo.”**



4. Under the SDS/FMS menu select “Generation Options.” Under the “Output” tab, the name of the output file name should be changed to “table.SQL.” If a Microsoft Access database is desired, the name of the database must be input under the Access tab.

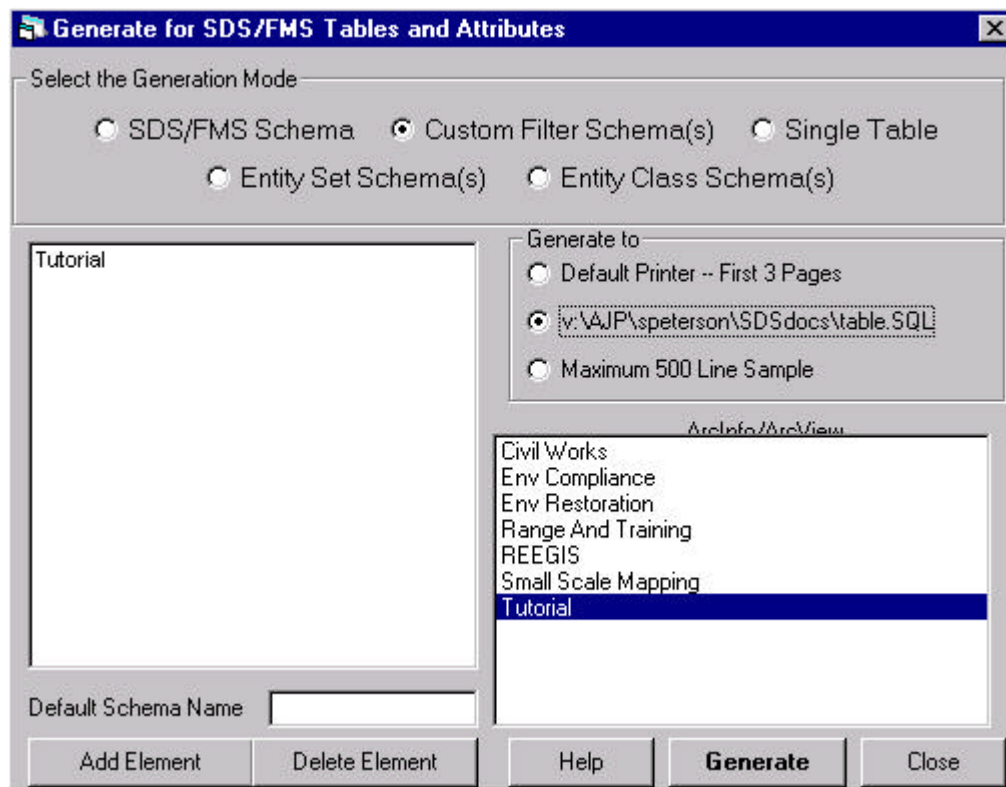




5. Select “Generate,” “New,” “Tables and Attributes.”

6. Select “Custom Filter Schema(s)”

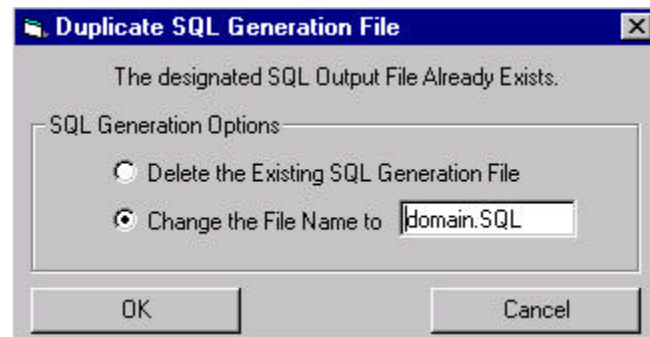
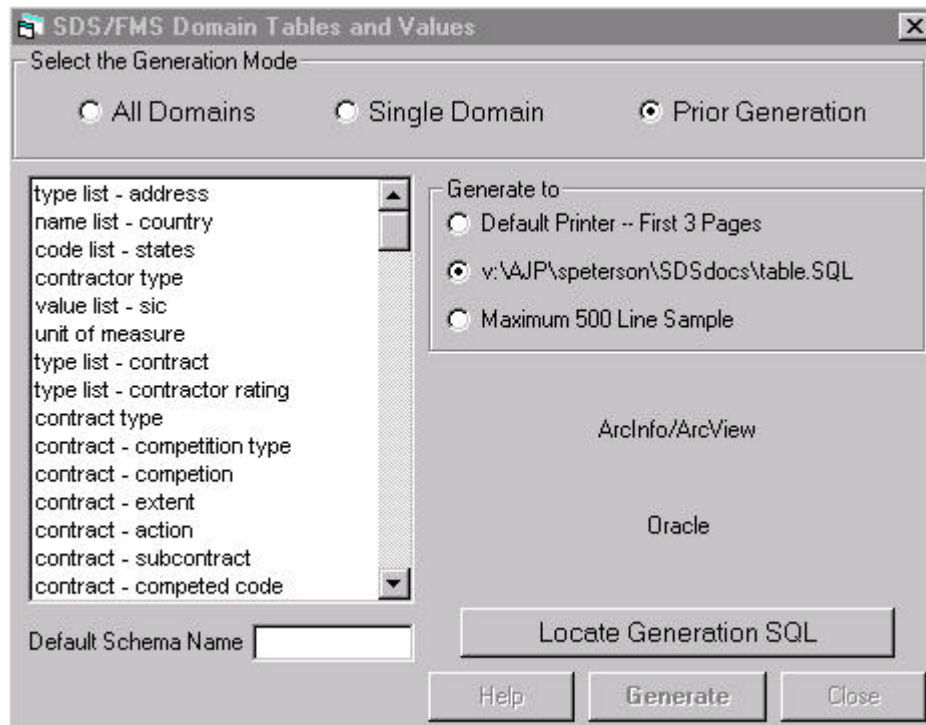
- “Tutorial” should be the last filter available in the list.
- Select “Add Element.” Tutorial should now be in the large box on the left of the screen.
- In the “Generate to” area, choose the filename you input as the Input File Name in 4.
- Click “Generate” and type “table.SQL” in the box next to “Change the File Name to.” Click “OK” and when generation is complete, click “Close.”



7. Select “Generate,” “New,” “Domains and Values.”

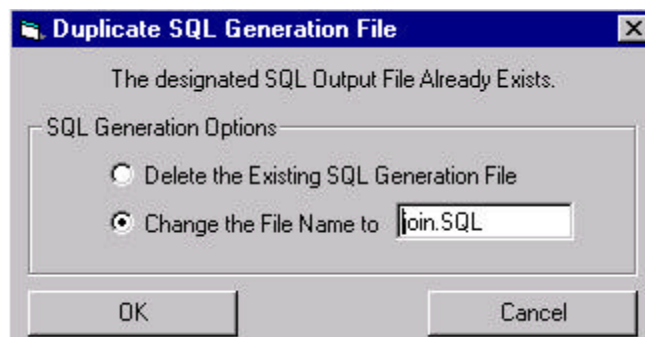
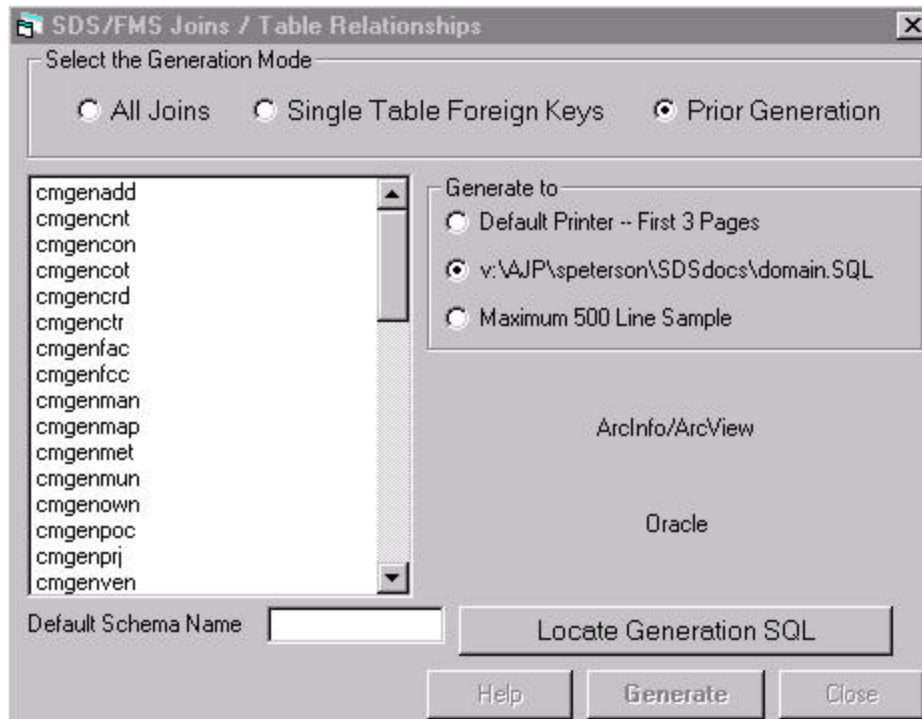
8. Select “Prior Generation”

- In the “Generate to” area, choose the filename you input as the Input File Name in 4.
- Click “Locate Generation SQL” and select “table.SQL.” Click “Open.”
- Click “Generate” and type “domain.SQL” in the box next to “Change the File Name to.” Click “OK” and when generation is complete, click “Close.”
- *Note: When using Microsoft Access, the user can create a separate database for the domains and values, but it is recommended that the user create a single database.



8. **Select “Generate,” “New,” “Relationships/Joins.”**
9. **Select “Prior Generation”**
 - In the “Generate to” area, choose the filename you input as the Input File Name in 4.
 - Click “Locate Generation SQL” and select “table.SQL.” Click “Open.”
 - Click “Generate” and type “join.SQL” in the box next to “Change the File Name to.” Click “OK” and when generation is complete, click “Close.”

- *Note: When using Microsoft Access as the DBMS, current limitations of Access limit the number of Joins that can be added to the database. Therefore, a narrative text file listing the links will be created and deposited into the home directory. These joins can be linked manually in Access but are not needed for implementation.

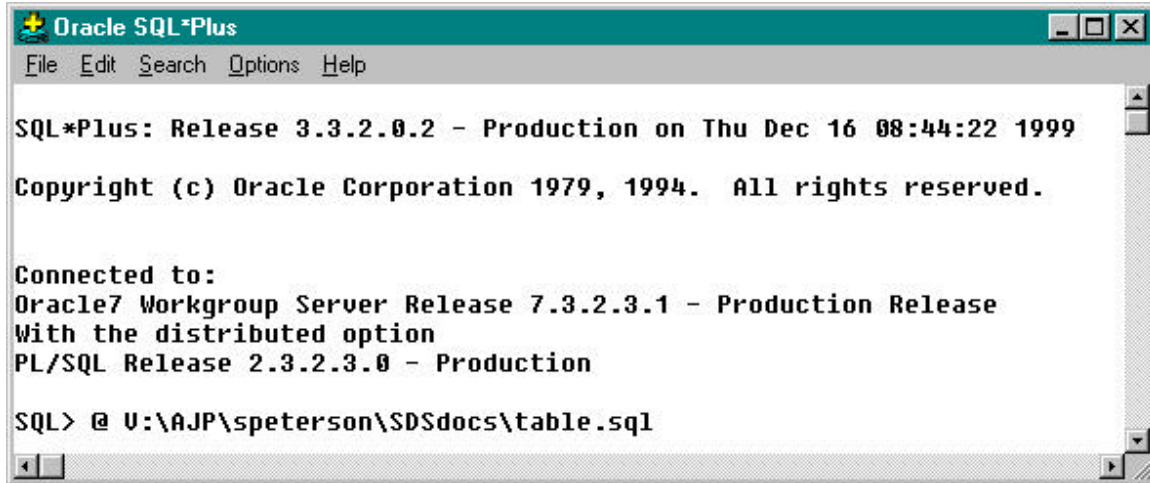


Database Population

Step 6. Database population is the next step in the implementation. This step will take the SQL statements created by the SDS/FMS Generator in the previous steps and create an empty Oracle database. Then, using the text file (lights.txt) provided with this tutorial, the user will populate the Oracle database with attributes:

1. **Connect to the Oracle database.**
2. **Using SQL Plus 3.3 (provided with Oracle) run the table.SQL that was created earlier in this tutorial.**

- Use the “@ table.SQL” command to run the SQL file.



```
Oracle SQL*Plus
File Edit Search Options Help

SQL*Plus: Release 3.3.2.0.2 - Production on Thu Dec 16 08:44:22 1999

Copyright (c) Oracle Corporation 1979, 1994. All rights reserved.

Connected to:
Oracle7 Workgroup Server Release 7.3.2.3.1 - Production Release
With the distributed option
PL/SQL Release 2.3.2.3.0 - Production

SQL> @ U:\AJP\speterson\SDSdocs\table.sql
```

3. Using SQL Plus 3.3 run the domain.SQL that was created earlier in this tutorial.



```
Oracle SQL*Plus
File Edit Search Options Help

SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL> @ U:\AJP\speterson\SDSdocs\domain.sql
```

4. Using SQL Plus 3.3 run the join.SQL that was created earlier in this tutorial.

- Note: Errors can be found after the SQL has been run; however they do not affect the process.



```
Oracle SQL*Plus
File Edit Search Options Help

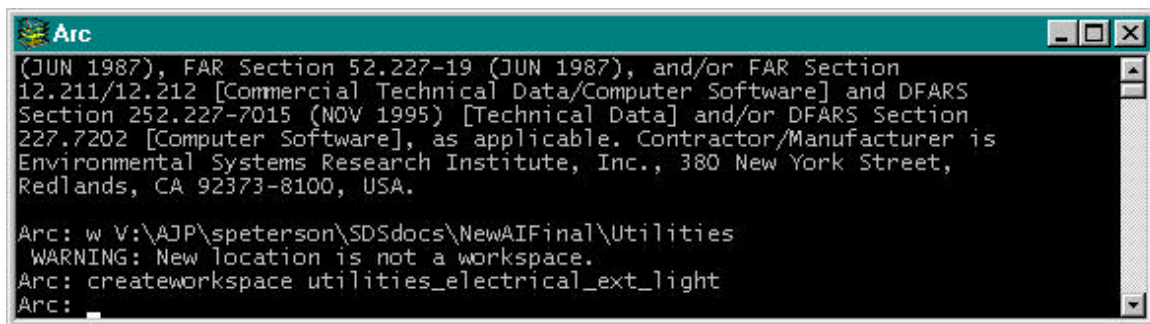
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL> @ U:\AJP\speterson\SDSdocs\join.sql
```

5. Using the lights.txt file for data comparison, use the standard SQL command “insert” for updating attribute information
 - Syntax: insert into (tablename) (the item names) values (the corresponding value to a new record in the item name) ---; semicolon needed for execution in Oracle
 - Example: “insert into utexllit (datalink, ext_lit_id, coord_x, coord_y, coord_z, lit_typ_d) values (251,' WP001', 2135528.94, 1022034.62, 50.10, 'STREET_LIGHT');”
 - Once this has been accomplished, run the lights.SQL in SQL Plus 3.3 to populate the Oracle database with the attribute information. Exit Oracle.

Database Integration

Step 7. The database integration step connects ArcInfo to the populated Oracle database, which in turn can be queried to provide information concerning the features.

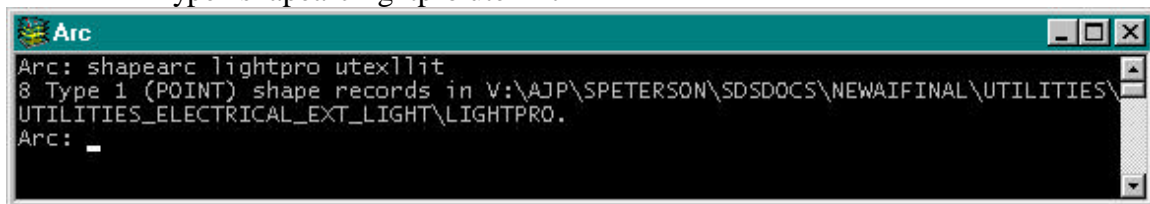
1. Create a directory entitled “Utilities” to store the data.
2. Start ArcInfo.
3. Using the “createworkspace” command, create a workspace named “utilities_electrical_ext_light” within the “utilities” directory to house the coverages for the utilities_electrical_ext_light entity class.



```
Arc
(JUN 1987), FAR Section 52.227-19 (JUN 1987), and/or FAR Section
12.211/12.212 [Commercial Technical Data/Computer Software] and DFARS
Section 252.227-7015 (NOV 1995) [Technical Data] and/or DFARS Section
227.7202 [Computer Software], as applicable. Contractor/Manufacturer is
Environmental Systems Research Institute, Inc., 380 New York Street,
Redlands, CA 92373-8100, USA.

Arc: w V:\AJP\speterson\SDSdocs\NewAIFinal\Utilities
WARNING: New location is not a workspace.
Arc: createworkspace utilities_electrical_ext_light
Arc:
```

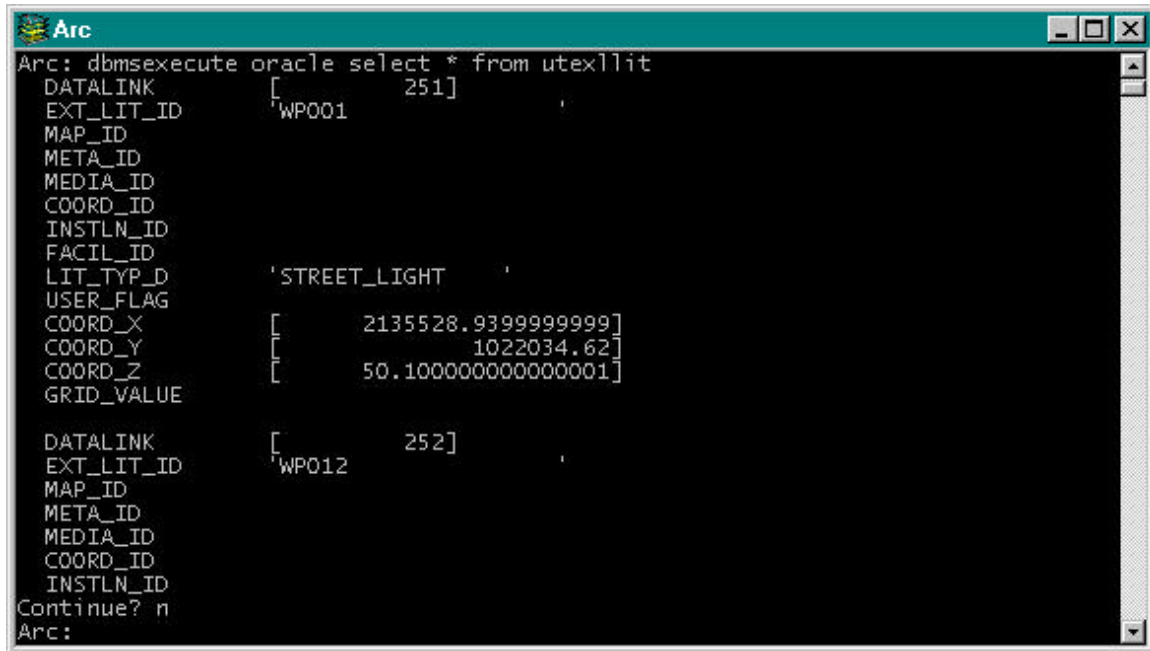
4. Using the “shapearc” command, import the shapefile “lightpro.shp” and convert it into an ArcInfo coverage named “utexllit”.
 - Type “shapearc lightpro utexllit”



```
Arc
Arc: shapearc lightpro utexllit
8 Type 1 (POINT) shape records in V:\AJP\SPETERSON\SDSDOCS\NEWAIFINAL\UTILITIES\
UTILITIES_ELECTRICAL_EXT_LIGHT\LIGHTPRO.
Arc:
```

5. Connect to the Oracle database using “show databases” and “connect”.
 - ”Show databases” shows the user the different types of databases to which ArcInfo can connect.
 - Type “connect oracle user/password@database alias”.
6. Validate connection by using “DBMSEXECUTE”.

- For example, “dbmsexecute oracle select * from utexllit”, which should list all of the Oracle items in the utexllit table.
- If tables are not listed, a step has been missed. Please refer to Step 1 of the Database Population section of this tutorial.



```

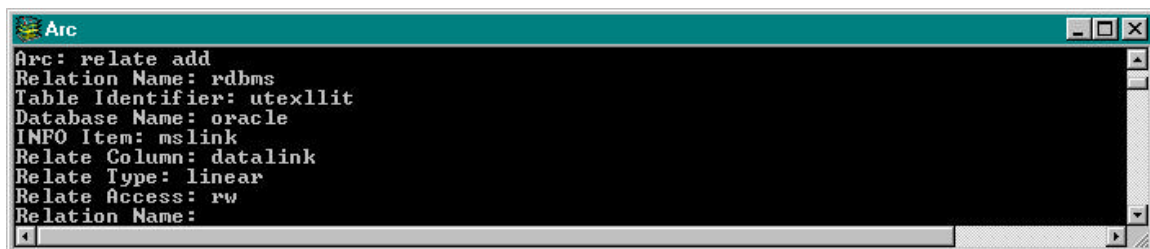
Arc: dbmsexecute oracle select * from utexllit
  DATALINK      [      251]
  EXT_LIT_ID    'WP001
  MAP_ID
  META_ID
  MEDIA_ID
  COORD_ID
  INSTLN_ID
  FACIL_ID
  LIT_TYP_D     'STREET_LIGHT
  USER_FLAG
  COORD_X       [      2135528.9399999999]
  COORD_Y       [      1022034.62]
  COORD_Z       [      50.100000000000001]
  GRID_VALUE

  DATALINK      [      252]
  EXT_LIT_ID    'WP012
  MAP_ID
  META_ID
  MEDIA_ID
  COORD_ID
  INSTLN_ID
Continue? n
Arc:

```

7. At the ‘Arc’ prompt, use the “Relate Add” command to relate the INFO and Oracle tables with “datalink” as the key. Use the following steps at the ArcInfo prompt to accomplish this:

- Arc: relate add
- Relation Name: RDBMS
- Table Identifier: utexllit
- Database: Oracle
- INFO Item: mslink
- Relate Column: datalink
- Relate Type: linear
- Relate Access: rw
- Press “Enter” key



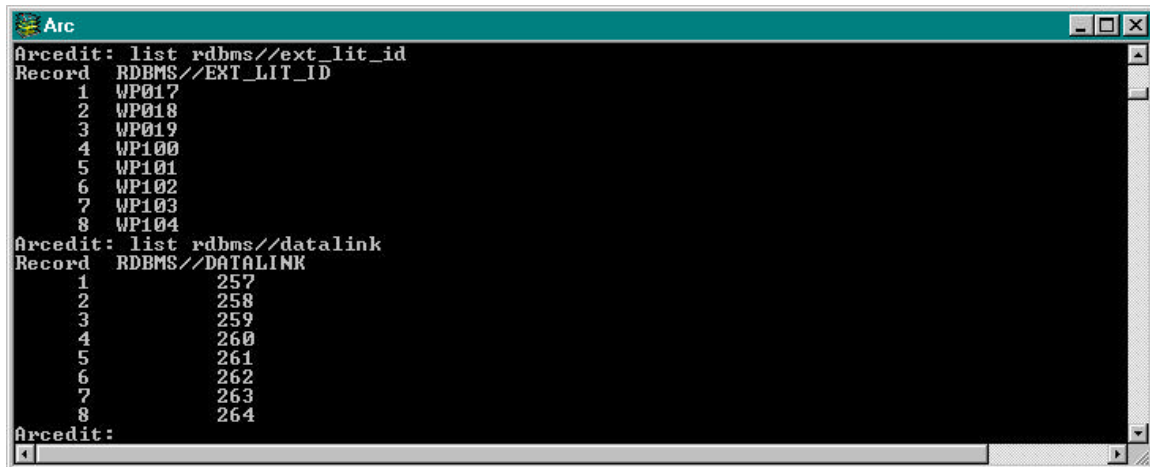
```

Arc: relate add
Relation Name: rdbms
Table Identifier: utexllit
Database Name: oracle
INFO Item: mslink
Relate Column: datalink
Relate Type: linear
Relate Access: rw
Relation Name:

```

8. The related data should be available for query using ARCPLOT, ARCEDIT, or TABLES.

- Open Arcedit
- Type "editcoverage utexllit"
- Type "editfeature labels"
- Type "select all"
- Type "list rdbms/(attribute) ext_lit_id"



```

Arcedit: list rdbms//ext_lit_id
Record  RDBMS//EXT_LIT_ID
1      WP017
2      WP018
3      WP019
4      WP100
5      WP101
6      WP102
7      WP103
8      WP104
Arcedit: list rdbms//datalink
Record  RDBMS//DATA LINK
1      257
2      258
3      259
4      260
5      261
6      262
7      263
8      264
Arcedit:

```

Bibliography

1. Groff, James R. and Weinberg, Paul N.; “Using SQL”; pp. 49-70; Osborne McGraw-Hill, 1990.